Claim amendments:

1. (Previously Presented) A method for transforming a datastream comprising the steps of:

[[a)]] parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, [[and]] wherein the plurality of work units are parsed from a single job, wherein each work unit may be either a data work unit or a control work unit, and wherein each control work unit may be an immediate control work unit or a scheduled control work unit or an interrupt control work unit;

queuing each data work unit on a queue accessible by a plurality of compute nodes;

queuing a scheduled control work unit at a tail of the queue to be processed by a compute node after all other work units presently in the queue;

queuing an immediate control work unit at a head of the queue to be processed by a compute node before all other work units in the queue;

forwarding an interrupt control work unit to a compute node immediately regardless of any work units in the queue; and

[[b)]] processing each of the plurality of work units by at least one compute node to convert each data work unit into a second format,

wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes.

2. (Currently Amended) The method of claim 1, wherein the parsing step [[(a)]] includes:

[[(a1)]] providing a plurality of sources, wherein each source is associated with at least one transform;

[[(a2)]] instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format; and

[[(a3)]] utilizing the at least one source to parse the datastream.

3. (Currently Amended) The method of claim 2, wherein the processing step [[(b)]] includes:

[[(b1)]] loading the at least one transform associated with the at least one instantiated source in the at least one compute node; and

[[(b2)]] utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format.

4. (Currently Amended) The method of claim 2 further comprising:

[[(c)]] load balancing the plurality of work units.

5. (Cancelled)

6. (Cancelled)

7. (Cancelled)

8. (Cancelled)

9. (Original) The method of claim 2, wherein the at least one source is instantiated as a dynamic library.

10. (Currently Amended) A computer readable medium containing program instructions for transforming a datastream, the program instructions for:

[[a)]] parsing the datastream into a plurality of work units in a first format wherein each work unit may be processed independent of all other work units, [[and]] wherein the plurality of work units are parsed from a single job, wherein each work unit may be either a data work unit or a control work unit, and wherein each control work unit may be an immediate control work unit or a scheduled control work unit or an interrupt control work unit;

queuing each data work unit on a queue accessible by a plurality of compute nodes;

queuing a scheduled control work unit at a tail of the queue to be processed by a compute node after all other work units presently in the queue;

queuing an immediate control work unit at a head of the queue to be processed by a compute node before all other work units in the queue;

forwarding an interrupt control work unit to a compute node immediately regardless of any work units in the queue; and

[[b)]] processing each of the plurality of work units by at least one compute node to convert each data work unit into a second format,

wherein the processing of each work unit is independent of processing of the other work units and wherein multiple work units are processed in parallel by multiple compute nodes.

11. (Currently Amended) The computer readable medium of claim 10, wherein the parsing instruction [[(a)]] includes:

[[(a1)]] providing a plurality of sources, wherein each source is associated with at least one transform;

[[(a2)]] instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format; and

[[a3)]] utilizing the at least one source to parse the datastream.

12. (Currently Amended) The computer readable medium of claim 11, wherein the processing instruction [[(b)]] includes:

[[(b1)]] loading the at least one transform associated with the at least one instantiated source in the at least one compute node; and

[[(b2)]] utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format.

13. (Currently Amended) The computer readable medium of claim 11 further comprising:

[[(c)]] load balancing the plurality of work units.

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Original) The computer readable medium of claim 11, wherein the at least one source is instantiated as a dynamic library.

19. (Currently Amended) A system for transforming a datastream comprising:

a central component for receiving the datastream in a first format;

a plurality of sources in the central component, wherein each of the plurality of sources is associated with at least one transform;

a queue; and

at least one compute node coupled to the central component via the queue,

wherein the central component instantiates at least one source of the plurality of sources that parses the datastream into a plurality of work units in the first format,

wherein each work unit may be processed independent of all other work units, [[and]]

wherein the plurality of work units are parsed from a single job, [[and]]

wherein the central component distributes each of the work units to the at least one compute node by queuing each data work unit on a queue accessible by a plurality of compute nodes, by queuing a scheduled control work unit at a tail of the queue to be processed by a compute node after all other work units presently in the queue, by queuing an immediate control work unit at a head of the queue to be processed by a compute node before all other work units in the queue, and by forwarding an interrupt control work unit to a compute node immediately regardless of any work units in the queue,

wherein the at least one compute node converts each data work unit into a second format independent of all other compute nodes operable on other work units, and

wherein at least two compute nodes are operable in parallel to convert at least two data work units in parallel.

20. (Original) The system of claim 19, wherein each of the at least one compute nodes loads the at least one transform as a dynamic library and utilizes the at least one transforms to convert a work unit in the first format to the second format.

21. (Currently Amended) The system of claim 19, wherein the central component further includes:

a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node~~,~~

~~wherein the load balancing mechanism generates at least one queue for the plurality of work units and dispatches each work unit from the at least one queue to the at least one compute node in an order received from the at least one source~~.

22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (Original) The system of claim 19, wherein the at least one source is instantiated as a dynamic library.